SPIKELITE: LIGHTWEIGHT SPIKING NEURAL NETWORKS FOR TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks (SNNs) offer an energy-efficient paradigm for time-series forecasting through spike-driven computation. However, recent SNN forecasters often pursue higher accuracy through increasingly complex attention mechanisms, or specialized neuronal dynamics, weakening the lightweight motivation of SNNs. We introduce SpikeLite, a spiking forecasting framework built around two modules: a Frequency-Selective Spiking Encoder (FSSE) for frequency-sensitive temporal encoding and a Sparse Spiking Channel Attention (SSCA) module for selective cross-channel interaction. FSSE exploits the low-pass filtering behavior of LIF dynamics to reorganize each input sequence into frequency-sensitive components while collectively preserving the input at the decomposition stage. SSCA then learns a binary mask from encoded channel representations and uses it to selectively exchange information within spike-driven self-attention, retaining informative cross-channel interactions while suppressing redundant ones. When explicit channel interaction is unnecessary, SpikeLite uses the lighter FSSE-only channel-independent path. Experiments under the SeqSNN and SpikF protocols cover four standard multivariate and eight long-term forecasting benchmarks. SpikeLite achieves the best aggregate performance under both protocols, with an average R^2 of 0.790 and RSE of 0.440, and lowest average MSE/MAE of 0.343/0.345 in long-term forecasting. Moreover, evaluation on the ECL dataset shows that SpikeLite achieves the lowest reported energy consumption, further demonstrating its potential for energy-efficient time-series forecasting.

1 INTRODUCTION

Spiking neural networks (SNNs) are regarded as the third generation of neural networks, emulate biological neuronal dynamics and communicate through discrete spike events, offering a biologically inspired and potentially energy-efficient alternative to artificial neural networks (ANNs) Maass (1997). Recent advances have extended SNNs to many tasks, including image classification Fang et al. (2021); Zhou et al. (2023), object detection Kim et al. (2020); Su et al. (2023), semantic segmentation Kim et al. (2022), speech recognition Wu et al. (2020); Pellegrini et al. (2021), and autonomous driving and control Zhu et al. (2024).

Multivariate time-series forecasting is an important predictive task with applications in energy systems, traffic management, weather prediction, and financial analysis Qiu et al. (2025); Han et al. (2024). Motivated by the energy-efficiency of SNNs, recent studies have begun to develop SNN-based forecasting models. SeqSNN was the first work to systematically apply SNNs to time-series forecasting, evaluating spiking convolutional, recurrent, and Transformer architectures within a unified forecasting framework Lv et al. (2024). Following this direction, SpikF introduces frequency-domain selection for long-term forecasting Wu et al. (2025), TS-LIF develops a dual-compartment spiking neuron to model multi-scale temporal dynamics Feng et al. (2025), and SpikeSTAG combines graph learning with spike-based temporal processing for spatio-temporal forecasting Hu et al. (2025). These advances improve the forecasting accuracy of SNNs, but increasingly complex neuronal dynamics, attention modules, and spatial–temporal backbones also add architectural and computational complexity, potentially compromising the energy-efficiency advantage that motivates spike-based modeling. This raises a central question: **how can SNN-based forecasters preserve their accuracy gains while maintaining energy-efficient computation?**

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We revisit this question through two fundamental aspects of multivariate forecasting: input-channel representation and cross-channel modeling. Existing spiking forecasters primarily focus on converting continuous observations into spike sequences, while making limited use of the latent characteristics embedded in the observed series. In particular, low-frequency variations often reflect slowly evolving trends, whereas higher-frequency variations capture rapid local changes and shorter-period patterns Wu et al. (2021); Zhou et al. (2022). At the same time, modeling relationships among variables requires more than dense interactions among all channels: useful information should be exchanged selectively while redundant channel connections are suppressed.

Based on these considerations, we introduce SpikeLite, a lightweight spiking framework for time-series forecasting that combines frequency-sensitive encoding with selective cross-channel interaction. Its Frequency-Selective Spiking Encoder (FSSE) uses parallel LIF branches with learnable decay factors to organize each input sequence into frequency-sensitive temporal components while collectively preserving the original signal. The Sparse Spiking Channel Attention (SSCA) module learns a binary channel-interaction mask to retain informative spike-driven exchanges and suppress redundant connections. We evaluate the model under the SeqSNN and SpikF protocols on four standard and eight long-term forecasting benchmarks, where SpikeLite achieves the strongest overall performance among spiking forecasters while requiring the lowest estimated energy consumption. Our contributions are summarized as follows:

- We introduce SpikeLite, a lightweight spiking forecasting framework that combines frequency-sensitive input representation with selective cross-channel interaction.
- We develop FSSE, which exploits heterogeneous LIF dynamics to construct frequency-sensitive components and aggregate them into compact channel representations, and SSCA, which uses a learned binary mask to suppress redundant spike-driven channel interactions.
- We establish comprehensive evaluations under two spiking forecasting protocols. It achieves the best aggregate accuracy among the compared ANN and SNN methods and the lowest estimated energy consumption in the reported energy comparison.

2 Related Work

2.1 Input-Channel Representation

Input-channel representation determines what information from the historical window is available for forecasting. Conventional models process historical observations through temporal convolutions, recurrent updates, or direct linear projections Lai et al. (2018); Bai et al. (2018); Zeng et al. (2023). To capture more complex temporal dependencies, other methods adopt elaborate Transformer backbones Wu et al. (2021); Zhou et al. (2022); Nie et al. (2023). Although deeper backbones improve modeling capacity, recent work shows that lightweight representations can remain highly competitive. LTSF-Linear directly maps the lookback window to the forecast horizon, while DLinear improves this design by separately projecting moving-average trends and residual components Zeng et al. (2023). Other methods organize the input through progressive decomposition and autocorrelation Wu et al. (2021), Fourier or wavelet representations Zhou et al. (2022), or local temporal patches Nie et al. (2023). However, such representations are designed for conventional continuous-valued networks rather than spiking models.

Spiking forecasters additionally need to encode continuous observations into discrete spike trains for spike-driven processing. Existing methods have successfully developed spiking representations of time-series inputs through temporal encoding and neuronal dynamicsLv et al. (2024). However, their primary focus is on how continuous observations are converted into effective spike sequences, while the intrinsic characteristics already present in the observed signals remain less explored. SpikeLite addresses this limitation through FSSE, which incorporates frequency characteristics into the spiking encoding process and produces frequency-sensitive representations for each input channel.

2.2 Cross-Channel Modeling

After individual channel representations are constructed, forecasting models differ in how they exchange information across variables. Channel-independent designs, such as DLinear and PatchTST, process each variable separately, often using a shared predictor structure or shared parameters across

channels Zeng et al. (2023); Nie et al. (2023). This strategy provides a simple and efficient prediction path, but may overlook useful cross-variable dependencies. Channel-dependent designs instead introduce explicit interactions among variables through convolution, graph propagation, or attention. For example, Crossformer uses two-stage attention to capture cross-dimension dependencies Zhang & Yan (2023), while iTransformer treats variables as tokens and applies self-attention to learn their multivariate correlations Liu et al. (2024).

Dense interaction, however, assumes that all channel pairs are equally useful and can propagate irrelevant information. Prior studies show that indiscriminate mixing may introduce noise or oversmoothing, motivating mechanisms that select informative relationships Chen et al. (2024); Huang et al. (2023). SNN-based forecasters also support cross-channel modeling: SeqSNN applies spiking self-attention to channel-wise embeddings Lv et al. (2024), and SpikeSTAG combines adaptive graph learning with spiking temporal processing for spatio-temporal forecasting Hu et al. (2025). These methods demonstrate the value of variable interaction, but their interaction structures are either dense or tied to specialized graph architectures. SpikeLite instead introduces SSCA, which learns a binary mask over channel pairs and uses it to retain informative spike-driven interactions while suppressing redundant connections.

3 PRELIMINARIES

3.1 PROBLEM FORMULATION

We first define the multivariate time-series forecasting task. Given a historical multivariate sequence $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_L]^\top \in \mathbb{R}^{L\times C}$, where L is the lookback length, C is the number of channels, and $\mathbf{x}_l \in \mathbb{R}^C$ contains all channel observations at position l, multivariate time-series forecasting aims to predict the next P observations, $\mathbf{Y} = [\mathbf{x}_{L+1}, \ldots, \mathbf{x}_{L+P}]^\top \in \mathbb{R}^{P\times C}$. A forecasting model f_θ with learnable parameters θ maps the historical sequence to the prediction $\widehat{\mathbf{Y}} = f_\theta(\mathbf{X})$. Throughout the paper, l indexes the original observation sequence, whereas $t \in \{1, \ldots, T_s\}$ indexes the internal simulation steps of a spiking layer.

3.2 SPIKING NEURAL COMPUTATION

Spiking neural networks (SNNs) model the evolution of neuronal membrane potentials and transmit information through discrete spike events. Among the various spiking neuron models, the leaky integrate-and-fire (LIF) neuron is widely adopted because it provides a simple mechanism for capturing temporal state dynamics. Given an input current $\mathbf{I}^t$ at simulation step t, an LIF neuron performs leaky integration, thresholding, and reset according to

$$\begin{cases} U^t = \lambda U^{t-1} + \mathbf{I}^t - \vartheta \mathbf{S}^{t-1}, \\ \mathbf{S}^t = \Theta\big(U^t - \vartheta\big), \\ \mathbf{G}^t = U^t \mathbf{S}^t. \end{cases} \tag{1}$$

Here, λ denotes the membrane decay factor. The last term implements a delayed subtractive reset. During training, the derivative of the Heaviside function is approximated with a surrogate gradient.

Based on these neuronal dynamics, spiking self-attention (SSA) extends pairwise representation interaction to spike-driven networks Zhou et al. (2023); Lv et al. (2024). Given an input representation $\mathbf{X}$, three learnable projections followed by spiking neurons produce binary query, key, and value representations,

$$\mathbf{Q} = \mathrm{SN}(\mathbf{X}\mathbf{W}_Q), \quad \mathbf{K} = \mathrm{SN}(\mathbf{X}\mathbf{W}_K), \quad \mathbf{V} = \mathrm{SN}(\mathbf{X}\mathbf{W}_V), \tag{2}$$

where $\mathrm{SN}(\cdot)$ denotes the spiking-neuron activation. SSA then aggregates pairwise interactions among these spike representations without relying on the softmax normalization used in conventional self-attention:

$$\mathrm{SSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{SN}\big(\alpha\, \mathbf{Q}\mathbf{K}^\top \mathbf{V}\big), \tag{3}$$

where α scales the magnitude of the aggregated interactions to maintain an appropriate activation range for the subsequent spiking neuron.

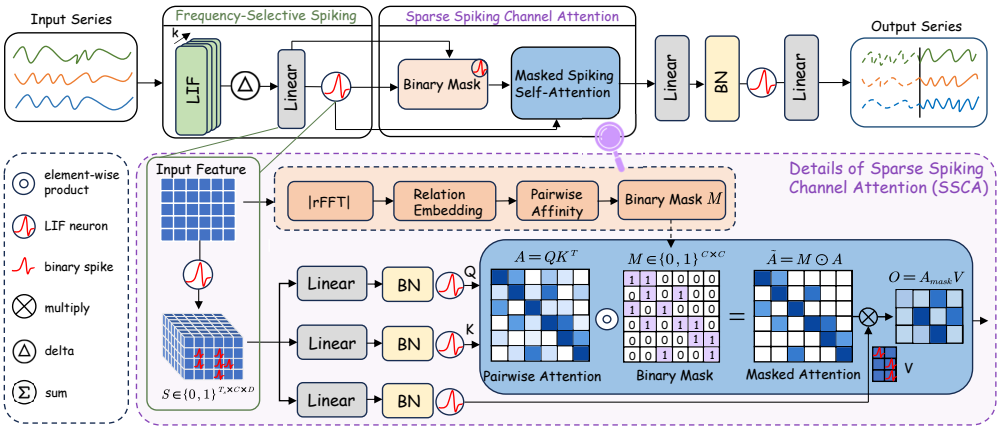


Figure 1: Overview of SpikeLite. FSSE constructs frequency-sensitive representations for individual input channels, while SSCA selectively exchanges information across channels. The lower panel details mask generation and masked spiking attention: the mask is estimated from the encoded channel representations and applied to the attention scores before value aggregation.

4 METHOD

4.1 OVERVIEW

As illustrated in Fig. 1, SpikeLite is a lightweight spiking framework for multivariate time-series forecasting that combines the Frequency-Selective Spiking Encoder (FSSE) with Sparse Spiking Channel Attention (SSCA). Given $\mathbf{X} \in \mathbb{R}^{L\times C}$, FSSE applies parallel LIF dynamics with learnable decay factors to each input channel, and reorganizes their event-gated responses into a compact representation $\mathbf{Z} \in \mathbb{R}^{C\times D}$, where $\mathbf{Z}_{c,:}$ represents the c-th channel. FSSE then repeats $\mathbf{Z}$ over T_s simulation steps and converts it into binary spike states $\mathbf{S}_{\text{in}}^{(t)} \in \{0,1\}^{C\times D}$. SSCA uses a learned binary interaction mask to selectively exchange information across channels while suppressing redundant connections; when omitted, the model follows the lighter FSSE-only path. A lightweight prediction head maps the resulting representation to $\widehat{\mathbf{Y}} = g_\psi(\widetilde{\mathbf{Z}}) \in \mathbb{R}^{P\times C}$.

4.2 FREQUENCY-SELECTIVE SPIKING ENCODER

Frequency-selective LIF dynamics. As defined in Section 3, the LIF neuron integrates the current input with its decayed membrane state. To analyze this dynamics in the frequency domain, we temporarily remove thresholding and reset, yielding the linearized equation $U_l = \tau U_{l-1} + X_l$. Its z-transform gives $H(z) = U(z)/X(z) = 1/(1-\tau z^{-1})$. Evaluating this transfer function on the unit circle, $z = e^{\mathrm{i}\omega}$, produces

$$H_\tau(e^{\mathrm{i}\omega}) = \frac{1}{1-\tau e^{-\mathrm{i}\omega}}, \qquad \left|H_\tau(e^{\mathrm{i}\omega})\right| = \frac{1}{\sqrt{1+\tau^2-2\tau\cos\omega}}, \tag{4}$$

where ω denotes the angular frequency. Thus, leaky integration behaves as a first-order low-pass filter: a larger τ more strongly favors low-frequency components, whereas a smaller τ produces a flatter response and retains a broader frequency range. Different decay factors therefore induce distinct frequency sensitivities to the same input, as illustrated in Fig. 2.

The complete LIF neuron restores thresholding and reset, making the response event-dependent. FSSE instantiates K parallel branches for each input variable, with branch-specific learnable decay factors. For branch k and variable c, the dynamics along the original observation axis are

$$U_{l,c}^{(k)} = \tau_c^{(k)} U_{l-1,c}^{(k)} + X_{l,c} - \vartheta_c^{(k)} S_{l-1,c}^{(k)}, \tag{5}$$

$$S_{l,c}^{(k)} = \Theta\left(U_{l,c}^{(k)} - \vartheta_c^{(k)}\right), \tag{6}$$

where $U_{l,c}^{(k)}$ is the membrane state, $S_{l,c}^{(k)} \in \{0,1\}$ is the emitted event, and $\vartheta_c^{(k)}$ is the firing threshold. Both $U_{0,c}^{(k)}$ and $S_{0,c}^{(k)}$ are initialized to zero. The last term in Eq.(5) implements the delayed subtractive reset.

FSSE gates each membrane response with its emitted spike: $G_{l,c}^{(k)} = U_{l,c}^{(k)} S_{l,c}^{(k)}$. Thus, $G_{l,c}^{(k)}$ is zero when branch k is inactive and retains the corresponding membrane state when the branch fires, combining the frequency-selective neuronal dynamics with event-based transmission.

Each decay factor is parameterized as $\tau_c^{(k)} = \sigma(\rho_c^{(k)})$, where $\rho_c^{(k)}$ is trainable and σ denotes the logistic sigmoid. We initialize the branches with descending decay factors to establish a nominal slow-to-fast ordering. During training, the decay factors are freely optimized without any explicit constraint on their relative ordering. Despite the absence of an explicit ordering constraint, the learned decay factors remain broadly consistent with the initial slow-to-fast ordering.

Frequency-sensitive encoding. To obtain informative frequency-sensitive components without discarding the original signal, FSSE applies successive differences to the event-gated responses of the parallel LIF branches. Let $\mathbf{G}^{(k)} \in \mathbb{R}^{L\times C}$ collect the responses of branch k over the historical window. FSSE constructs $K+1$ components as

$$\mathbf{B}^{(j)} = \begin{cases} \mathbf{G}^{(1)}, & j = 0, \\ \mathbf{G}^{(j+1)} - \mathbf{G}^{(j)}, & 1 \le j < K, \\ \mathbf{X} - \mathbf{G}^{(K)}, & j = K. \end{cases} \tag{7}$$

The first component retains the response of the first LIF branch, the intermediate components describe the response differences between adjacent branches, and the final component preserves the residual between the input and the last branch response. Because neighboring branch responses are subtracted successively, the components form a telescoping decomposition:

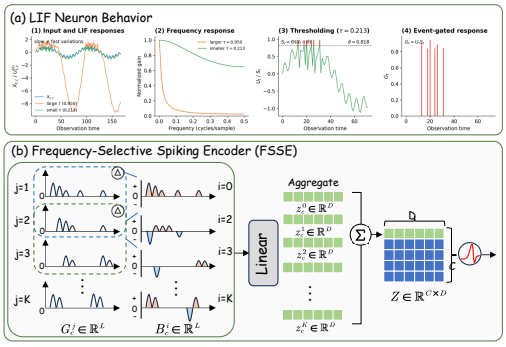


Figure 2: LIF responses and the FSSE encoding process. (a) LIF responses under different decay factors. (b) Details of Frequency-Selective Spiking Encoder.

$$\sum_{j=0}^{K} \mathbf{B}^{(j)} = \mathbf{G}^{(1)} + \sum_{j=1}^{K-1} \left(\mathbf{G}^{(j+1)} - \mathbf{G}^{(j)}\right) + \left(\mathbf{X} - \mathbf{G}^{(K)}\right) = \mathbf{X}. \tag{8}$$

The decomposition in Eq.(8) produces the components $\{\mathbf{B}^{(j)}\}_{j=0}^{K}$ while preserving the input collectively. Each component is independently projected along the observation axis and aggregated into a compact channel representation:

$$\begin{aligned} \mathbf{z}_c^{(j)} &= \mathbf{W}^{(j)} \mathbf{B}_{:,c}^{(j)} + \mathbf{b}^{(j)} \in \mathbb{R}^D, \\ \mathbf{Z}_{c,:} &= \sum_{j=0}^{K} \alpha_j \mathbf{z}_c^{(j)}, \qquad \mathbf{Z} \in \mathbb{R}^{C\times D}, \end{aligned} \tag{9}$$

where $\mathbf{W}^{(j)} \in \mathbb{R}^{D\times L}$ and α_j is a learnable scale for component j. The resulting $\mathbf{Z}$ is passed to the spike-driven interaction stage described in Section 4.3.

4.3 Sparse Spiking Channel Attention

FSSE produces frequency-sensitive representations for each input variable while collectively preserving the original signal. Because forecasting can also depend on relationships among variables, SSCA selectively exchanges information across these representations. A standard spiking self-attention block considers all channel pairs, whereas SSCA learns a sample-dependent binary mask $\mathbf{M} \in \{0,1\}^{C\times C}$ to retain informative connections during spike-driven aggregation and suppress redundant interactions.

Binary mask generation. Before spike replication, SSCA applies a real-valued Fourier transform to each row of $\mathbf{Z}$ along the representation dimension:

$$\mathbf{F}_c = |\mathrm{rFFT}\left(\mathbf{Z}_{c,:}\right)| \in \mathbb{R}^F, \qquad F = \left\lfloor \frac{D}{2} \right\rfloor + 1. \tag{10}$$

These descriptors are used only to estimate the binary channel-interaction mask. The subsequent attention operation is performed on the spike-form channel states obtained after replicating $\mathbf{Z}$ over the T_s simulation steps, rather than on the spectral descriptors.

To measure the similarity between channel descriptors, SSCA maps each spectral descriptor into a low-dimensional relation space using a trainable projection matrix:

$$\mathbf{r}_c = \mathbf{W}_r \mathbf{F}_c \in \mathbb{R}^r, \qquad \mathbf{W}_r \in \mathbb{R}^{r \times F}. \tag{11}$$

For a pair of channels (i, j), we compute the squared relation distance and its inverse affinity:

$$d_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|_2^2 + \epsilon, \qquad a_{ij} = d_{ij}^{-1}. \tag{12}$$

The affinity is normalized independently for each source channel:

$$p_{ij} = \mathrm{clip}\left(\gamma \frac{a_{ij}}{\max_{q \neq i} a_{iq}}, 0, 1\right), \qquad i \neq j. \tag{13}$$

Channels with similar spectral descriptors therefore obtain larger affinity values and are more likely to remain connected.

We obtain the binary interaction mask by thresholding the normalized affinity while retaining all self-connections:

$$\overline{m}_{ij} = \begin{cases} 1, & i = j, \\ \mathbb{I}(p_{ij} > \eta), & i \neq j. \end{cases} \tag{14}$$

To optimize this discrete mask, we use a straight-through estimator:

$$m_{ij} = \begin{cases} 1, & i = j, \\ \mathrm{sg}\left(\overline{m}_{ij} - p_{ij}\right) + p_{ij}, & i \neq j, \end{cases} \qquad \mathbf{M} = [m_{ij}], \tag{15}$$

where η is the sparsification threshold and $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. The forward value of $\mathbf{M}$ is binary, while its backward gradient is propagated through the continuous affinity p_{ij}.

Masked spike-driven attention. The spike-form variable states produced by FSSE are used as the input tokens of sparse SSA. For attention head m and simulation step t, the corresponding queries, keys, and values are denoted by $\mathbf{Q}^{(t,m)}$, $\mathbf{K}^{(t,m)}$, and $\mathbf{V}^{(t,m)}$. SSCA applies the learned mask directly to the pairwise interaction matrix:

$$\mathbf{A}_{\mathrm{SSCA}}^{(t,m)} = \kappa\left(\mathbf{Q}^{(t,m)} \mathbf{K}^{(t,m)\top}\right) \odot \mathbf{M}, \qquad \mathbf{O}^{(t,m)} = \mathbf{A}_{\mathrm{SSCA}}^{(t,m)} \mathbf{V}^{(t,m)}. \tag{16}$$

where κ is the attention scaling factor. Consistent with spike-driven SSA, no softmax normalization is applied. A zero entry $M_{ij} = 0$ removes the contribution from variable j to variable i, whereas $M_{ij} = 1$ preserves the corresponding interaction.

The masked outputs across all attention heads and simulation steps are aggregated through the SSA output projection, yielding the channel-interaction-enhanced representation $\widetilde{\mathbf{Z}}$. This representation is subsequently mapped by the prediction head to the final forecast $\widehat{\mathbf{Y}} \in \mathbb{R}^{P \times C}$.

5 Experiments

5.1 Experimental Settings

We evaluate SpikeLite under two representative forecasting settings: the standard multivariate forecasting protocol introduced by SeqSNN Lv et al. (2024) and the long-term forecasting protocol adopted by SpikF Wu et al. (2025). Across the two settings, we compare SpikeLite with a broad collection of ANN- and SNN-based forecasting models on datasets covering traffic, electricity, weather, and other application domains.

Table 1: Experimental results on standard multivariate forecasting under the SeqSNN protocol Lv et al. (2024). **Bold** and underlined values denote the best and second-best results, respectively. ↑/↓ indicates that higher/lower values are better.

Method	Metric	METR-LA				PEMS-BAY				Solar				Electricity				Summary	
		6	24	48	96	6	24	48	96	6	24	48	96	6	24	48	96	Avg.	Rank
ARIMA	R^2 ↑	.687	.441	.282	.265	.741	.723	.692	.670	.951	.847	.725	.689	.963	.960	.914	.863	.713	9.69
	RSE↓	.575	.742	.889	.902	.532	.548	.562	.612	.202	.365	.588	.589	.522	.534	.564	.599	.583	9.44
GP	R^2 ↑	.685	.437	.265	.233	.732	.712	.689	.665	.944	.836	.711	.675	.962	.968	.912	.852	.705	11.03
	RSE↓	.572	.738	.912	.925	.544	.532	.577	.592	.225	.388	.612	.575	.603	.612	.633	.642	.605	10.28
Autoformer	R^2 ↑	.762	.548	.411	.282	.782	.711	.689	.668	.960	.852	.791	.701	.980	.977	.975	.963	.753	8.00
	RSE↓	.569	.692	.785	.872	.452	.543	.577	**.565**	.212	.432	.622	.685	.481	.506	.566	.548	.569	8.91
PatchTST	R^2 ↑	.819	.618	.434	.284	.873	.718	.686	.612	.962	.868	.795	.729	.980	.977	.975	.963	.768	6.50
	RSE↓	.456	.666	.792	.875	.385	.532	.569	.577	.201	.369	.467	.588	.264	.325	.326	.441	.490	6.41
AGCRN	R^2 ↑	.764	.562	.442	.287	.787	.709	.694	.659	.958	.854	.788	.704	.987	.982	.978	.966	.758	6.66
	RSE↓	.511	.696	.785	.874	.543	.543	.566	.574	.207	.432	.625	.656	.223	.265	.296	.531	.520	7.56
STAEformer	R^2 ↑	.774	.573	.396	.279	.803	.716	.691	.647	.961	.857	.790	.713	.977	.973	.970	.959	.755	8.38
	RSE↓	.499	.686	.817	.881	.397	.534	.565	.580	.210	.431	.621	.601	.271	.303	.326	.481	.513	7.81
iTransformer	R^2 ↑	.829	**.623**	.439	.285	.887	.719	.685	.668	**.964**	.879	.799	.738	.979	.977	.975	.964	.776	4.84
	RSE↓	.436	**.648**	.780	.878	.362	.547	.561	.584	.191	.348	.448	.563	.259	.305	.335	.427	.479	5.09
TS-TCN	R^2 ↑	.810	.605	**.473**	.328	**.897**	.759	.698	.652	**.964**	.884	.762	.720	.980	.971	.968	.962	.777	5.41
	RSE↓	.459	.656	.757	.857	**.354**	.527	**.559**	.633	**.189**	**.325**	.484	.523	.264	.316	.318	.360	.474	4.22
TS-GRU	R^2 ↑	**.848**	.618	.430	**.329**	.874	.742	.684	.649	.938	.878	.779	.722	.991	.981	.983	.976	.776	5.47
	RSE↓	**.412**	.651	.795	**.853**	.384	.530	.587	.637	.253	.349	.426	.527	.216	.240	.236	.271	.460	5.19
iSpikformer	R^2 ↑	.817	.618	.440	.279	.879	.744	.687	**.674**	.961	.876	.795	.738	.977	.974	.972	.963	.775	5.88
	RSE↓	.475	.668	**.752**	.905	.376	.536	.569	.580	.204	.333	.465	.521	.263	.284	.338	.348	.476	5.56
TS-former	R^2 ↑	.847	.620	.445	.283	.874	.735	.683	.669	.961	**.886**	**.828**	**.774**	.987	.985	.981	.977	.783	3.78
	RSE↓	.416	.655	.783	.874	.379	.539	.572	.583	.224	.331	**.382**	**.435**	.197	.215	.234	.261	.443	4.16
SpikeLite	R^2 ↑	.847	.622	.452	.316	.889	**.781**	**.715**	.639	.962	.885	.811	.767	**.993**	**.991**	**.988**	**.983**	**.790**	**2.38**
	RSE↓	.414	.649	.781	.873	.361	**.506**	.577	.649	.201	.348	.446	.496	**.146**	**.167**	**.196**	**.233**	**.440**	**3.38**

Protocols and metrics. Under the SeqSNN protocol Lv et al. (2024), we evaluate standard multivariate forecasting with prediction horizons of 6, 24, 48, and 96, using R^2 and elative squared error (RSE) as evaluation metrics. Under the SpikF protocol Wu et al. (2025), we evaluate long-term forecasting with a lookback length of 96 and prediction horizons of 96, 192, 336, and 720, using mean squared error (MSE) and mean absolute error (MAE). Main-text results are averaged over the four prediction horizons, while the complete horizon-wise results are provided in Appendix B. Dataset statistics and complete experimental configurations are summarized in Appendix A.

Compared methods. For the SeqSNN protocol, we compare against statistical predictors, including ARIMA and Gaussian Process regression, and ANN forecasters, including Autoformer, PatchTST, and iTransformer. ARIMA, Gaussian Process, Autoformer, iTransformer, and iSpikformer results are taken from SeqSNN Lv et al. (2024). TS-TCN, TS-GRU, and TS-former results are taken from TS-LIF Feng et al. (2025). PatchTST, AGCRN, and STAEformer are implemented and evaluated under the SeqSNN data splits and evaluation procedure, using their original architectures Nie et al. (2023); Bai et al. (2020); Liu et al. (2023).

For the SpikF protocol (long-term forecasting), we compare SpikeLite with the ANN baselines iTransformer, RLinear, PatchTST, Crossformer, TimesNet, DLinear, SCINet, and Autoformer, together with SpikF. Their reported results are taken from the SpikF benchmark Wu et al. (2025), while SpikeLite is evaluated under the same protocol.

Table 2: Long-term forecasting performance under the SpikF protocol Wu et al. (2025). Results are averaged across four prediction horizons (96, 192, 336, and 720) and three random seeds. Lower MSE and MAE indicate better performance. **Bold** and underlined values indicate the best and second-best results, respectively.

Dataset	iTransformer		RLinear		PatchTST		Crossformer		TimesNet		DLinear		SCINet		Autoformer		**SpikF**		**SpikeLite**	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	.178	.270	.219	.298	.205	.290	.244	.334	.192	.295	.212	.300	.268	.365	.227	.338	.183	.275	**.175**	**.263**
Weather	.258	.278	.272	.291	.259	.281	.259	.315	.259	.287	.265	.317	.292	.363	.338	.382	**.245**	**.265**	**.245**	**.265**
ETTh1	.454	.447	.446	.434	.469	.454	.529	.522	.458	.450	.456	.452	.747	.647	.496	.487	**.440**	**.428**	.443	.429
ETTh2	.383	.407	.374	.398	.387	.407	.942	.684	.414	.427	.559	.515	.954	.723	.450	.459	**.372**	**.394**	.374	**.394**
ETTm1	.407	.410	.414	.407	**.387**	.400	.513	.496	.400	.406	.403	.407	.485	.481	.588	.517	.388	**.385**	**.387**	.389
ETTm2	.288	.332	.286	.327	.281	.326	.757	.610	.291	.333	.350	.401	.571	.537	.327	.371	.281	.320	**.278**	**.319**
Traffic	**.428**	**.282**	.626	.378	.481	.304	.550	.304	.620	.336	.625	.383	.804	.509	.628	.379	.497	.296	.479	.294
Exchange	.360	.403	**.354**	.414	.367	.404	.940	.707	.416	.443	**.354**	.414	.750	.626	.613	.539	.360	**.402**	.366	.404
Avg.	.345	.354	.374	.368	.355	.358	.592	.497	.381	.372	.403	.399	.609	.531	.458	.434	.346	.346	**.343**	**.345**
Avg. Rank	3.56	3.56	5.12	5.00	4.38	4.19	8.25	8.31	5.50	5.50	5.94	7.12	9.38	9.38	8.38	8.38	2.44	**1.75**	**2.06**	1.81

5.2 Main Results

SpikeLite achieves the best overall forecasting performance under both the standard multivariate and long-term forecasting settings.

Under the **SeqSNN protocol**, SpikeLite achieves the best overall performance, with an average R^2 of 0.790 and an average RSE of 0.440. Across the 16 dataset–horizon settings for each metric, SpikeLite obtains the largest number of first- and second-place results.

Among the four benchmarks, the clearest improvement appears on Electricity, whose pronounced periodic structure is well aligned with FSSE's reorganization of the input into multi-scale frequency-sensitive components. SpikeLite also performs strongly on METR-LA and PEMS-BAY, where correlated sensor measurements require both temporal modeling and information exchange across variables. Among the compared methods, TS-former ranks second and shows a slight advantage over SpikeLite on Solar, possibly due to its stronger emphasis on local temporal dynamics.

Under the **SpikF protocol**, SpikeLite achieves the best average MSE and MAE, demonstrating that its performance advantage also holds under the long-term forecasting setting.

SpikeLite extends a consistent advantage on Electricity to ECL, substantially outperforming all compared methods in long-term forecasting. Across the ETT and Traffic benchmarks, SpikeLite remains competitive with the strongest ANN and SNN baselines. The ETT datasets exhibit multi-scale temperature and load variations, whereas Traffic contains recurring patterns among spatially correlated sensor measurements; FSSE and SSCA jointly model these temporal and cross-channel characteristics. Exchange is more challenging due to its less regular long-horizon variations, yet SpikeLite remains close to the best-performing methods.

5.3 Frequency-Selective Mechanism Analysis

We further examine whether the parallel LIF branches and the resulting FSSE components exhibit distinct responses to temporal frequencies. Following the frequency-response analysis in Section 4.2, we apply sinusoidal inputs to the PEMS setting and sweep the frequency from 0.1 to 0.5 cycles per sample. The higher-frequency portion of this sweep provides a clearer view of the differences among the learned responses, as shown in Fig. 3.

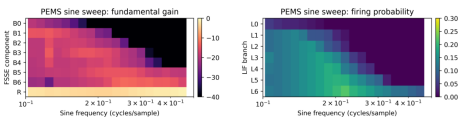


Figure 3: Frequency-sweep analysis of FSSE: component-wise fundamental gain and branch-wise firing probability.

In Fig. 3(a), the FSSE components exhibit distinct gain profiles: earlier components respond more strongly at lower frequencies, whereas later components and the residual remain responsive over a broader frequency range. Fig. 3(b) shows a similar overall trend in firing probability. The

firing-probability profiles broadly follow the nominal slow-to-fast initialization order, while allowing channel-dependent deviations after training. These results are consistent with the heterogeneous decay factors learned by FSSE and demonstrate that heterogeneous LIF dynamics and event gating produce distinct frequency-sensitive responses.

5.4 Ablation Study

To assess the contributions of FSSE and SSCA, we perform an ablation study under the SpikF protocol. The main-text analysis focuses on Traffic, where dependencies among observed variables make cross-channel modeling particularly relevant. The full model is compared against *w/o FSSE* and *w/o SSCA* with all other components and training settings unchanged. Figure 4 reports the Traffic results, with the corresponding Weather results provided in Appendix D.

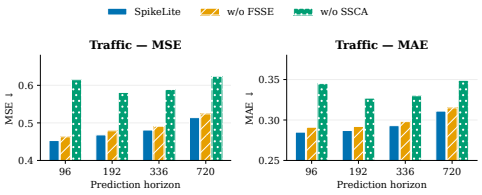


Figure 4: Ablation results on Traffic under the SpikF protocol. Lower values indicate better forecasting performance.

Removing either component degrades performance across all four Traffic horizons. Without FSSE, the average MSE and MAE increase from 0.479 and 0.294 to 0.489 and 0.299, indicating that frequency-sensitive encoding improves the representation of each variable's temporal variations. Removing SSCA produces a larger increase, to 0.602 MSE and 0.338 MAE, showing that selective cross-channel interaction contributes substantial predictive information on Traffic. Together, these results support the complementary roles of FSSE in temporal encoding and SSCA in modeling dependencies among variables.

5.5 Energy Analysis

Following the established energy-estimation protocol based on 45-nm CMOS technology Horowitz (2014), where a multiply–accumulate operation is assigned an energy cost of 4.6 pJ and an accumulate operation is assigned 0.9 pJ, we estimate the inference cost of the encoder backbone. The task-specific prediction heads are excluded for all methods to ensure a consistent comparison.

Table 3 shows that SpikeLite uses only 67.2K parameters and 0.02G operations, corresponding to an estimated energy consumption of $95.30\,\mu$J per sample, which is the lowest among the compared models. Compared with SpikF, SpikeLite reduces energy consumption by approximately 19.0%, while the savings over the ANN-based baselines are approximately 53.6% relative to DLinear and 97.1% relative to iTransformer.

Table 3: Model complexity, estimated energy, and forecasting accuracy on ECL at a prediction horizon of 720.

Model	Params	OPs	Energy (μJ)	MSE@720
SpikF	1.2K	0.13G	117.66	.219
DLinear	0.14M	45M	205.19	.245
iTransformer	1.6M	0.72G	3289.29	.225
SpikeLite	67.2K	0.02G	95.30	.218

These results indicate that the energy advantage of SpikeLite is attributed to its spike-driven encoder and compact channel interaction pathway, which reduce the number of costly operations required during inference while maintaining forecasting accuracy.

6 Conclusion

We presented SpikeLite, a lightweight spiking framework for multivariate time-series forecasting. FSSE uses heterogeneous LIF dynamics to reorganize each input sequence into frequency-sensitive components, while SSCA selectively models informative cross-channel interactions through a sparse binary mask. Experiments under the SeqSNN and SpikF protocols show that SpikeLite achieves consistently strong forecasting accuracy across standard and long-term benchmarks, while requiring the lowest estimated energy among the compared models. The limitations and future directions are discussed in Appendix H.

7 Reproducibility Statement

The supplementary material documents the dataset characteristics, temporal properties, preprocessing and evaluation protocols, and implementation and training configurations (Appendix A). It also provides the complete horizon-wise results, seed-stability analysis, ablations, frequency diagnostics, and energy-accounting details. Results produced by our implementation are averaged over three independent random seeds unless stated otherwise. All corresponding mean$\pm$std results are reported in Appendix C. Numerical results copied from published benchmark reports are identified in the corresponding comparison description and are retained under their original evaluation protocol; they are not presented as new three-seed measurements. The source code, configuration files, and analysis scripts will be released upon publication.

8 Ethics Statement

Our experiments use publicly available multivariate time-series benchmarks covering traffic sensors, electricity and solar measurements, weather observations, transformer measurements, and exchange rates. We collect no new human or animal data, and our experiments do not use individual-level identifiers or attempt to infer personal attributes. The datasets are used for research in accordance with their original licenses and documentation. Although aggregate traffic, energy, and exchange rates can support sensitive operational or mobility-related inferences in downstream systems, such deployments should follow the applicable data-use restrictions, privacy requirements, and access-control practices.

References

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Jialin Chen, Jan Eric Lenssen, Aosong Feng, Weihua Hu, Matthias Fey, Leandros Tassiulas, Jure Leskovec, and Rex Ying. From similarity to superiority: Channel clustering for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21056–21069, 2021.

Shibo Feng, Wanjin Feng, Xingyu Gao, Peilin Zhao, and Zhiqi Shen. TS-LIF: A temporal segment spiking neuron network for time series forecasting. In *International Conference on Learning Representations*, 2025.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. SOFTS: Efficient multivariate time series forecasting with series-core fusion. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2046.

Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10–14. IEEE, 2014. doi: 10.1109/ISSCC.2014.6757323.

Bang Hu, Changze Lv, Mingjie Li, Yunpeng Liu, Xiaoqing Zheng, Fengzhe Zhang, Wei Cao, and Fan Zhang. SpikeSTAG: Spatial-temporal forecasting via GNN-SNN collaboration. *arXiv preprint arXiv:2508.02069*, 2025.

Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. Crossgnn: Confronting noisy multivariate time series via cross interaction refinement. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46885–46902, 2023.

Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: Spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11270–11277, 2020. doi: 10.1609/aaai.v34i07.6787.

Youngeun Kim, Joshua Chough, and Priyadarshini Panda. Beyond classification: Directly training spiking neural networks for semantic segmentation. *Neuromorphic Computing and Engineering*, 2(4):044015, 2022. doi: 10.1088/2634-4386/ac9b86.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 95–104. ACM, 2018. doi: 10.1145/3209978.3210006.

Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 4125–4129, 2023. doi: 10.1145/3583780.3615160.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024.

Changze Lv, Yansen Wang, Dongqi Han, Xiaoqing Zheng, Xuanjing Huang, and Dongsheng Li. Efficient and effective time-series forecasting with spiking neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 33624–33637, 2024.

Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997. doi: 10.1016/S0893-6080(97)00011-7.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Thomas Pellegrini, Romain Zimmer, and Timothée Masquelier. Low-activity supervised convolutional spiking neural networks applied to speech commands recognition. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 97–103. IEEE, 2021. doi: 10.1109/SLT48900.2021.9383587.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. DUET: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1185–1196, 2025. doi: 10.1145/3690624.3709325.

Qiaoyi Su, Yuhong Chou, Yifan Hu, Jianing Li, Shijie Mei, Ziyang Zhang, and Guoqi Li. Deep directly-trained spiking neural networks for object detection. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6555–6565. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00603.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021.

Jibin Wu, Emre Yılmaz, Malu Zhang, Haizhou Li, and Kay Chen Tan. Deep spiking neural networks for large vocabulary automatic speech recognition. *Frontiers in Neuroscience*, 14:199, 2020. doi: 10.3389/fnins.2020.00199.

Wenjie Wu, Dexuan Huo, and Hong Chen. SpikF: Spiking fourier network for efficient long-term prediction. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67342–67368. PMLR, 2025.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 2022.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=frE4fUwz_h.

Rui-Jie Zhu, Ziqing Wang, Leilani Gilpin, and Jason K. Eshraghian. Autonomous driving with spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4346.

A Experimental Details

This appendix records the dataset characteristics, evaluation rules, implementation choices, additional forecasting results, and diagnostic experiments used in the paper. The two protocols are kept separate because they use different datasets, lookback windows, horizons, metrics, and model interfaces.

A.1 Datasets and temporal characteristics

Table 4 lists the dimensionality exposed to the forecasting model. The sensor-network datasets contain substantial cross-variable structure, whereas the physical and financial datasets are more heterogeneous in their temporal regularity. All datasets used here are regularly sampled benchmark series with complete input windows; explicit missing-value handling is not part of the current pipeline.

Table 4: Datasets used by the two evaluation protocols. "Variables/nodes" denotes the input dimensionality after protocol preprocessing.

Protocol	Dataset	Domain	Variables/nodes	Main temporal characteristics
SeqSNN	METR-LA	traffic sensors	207	Road-speed measurements with spatial correlation, daily repetition, and short-term congestion changes.
	PEMS-BAY	traffic sensors	325	Large sensor network with correlated locations, recurring traffic cycles, and rapidly changing local conditions.
	Solar	solar power	137	Diurnal structure with weather-driven local fluctuations and intermittent high-frequency changes.
	Electricity	electricity load	321	Heterogeneous consumption channels with periodic structure and channel-specific temporal profiles.
SpikF	ECL	electricity load	321	Long-horizon electricity demand with periodic and heterogeneous channel behavior.
	Weather	meteorological series	21	Smooth physical variables with seasonal trends and multi-scale local variation.
	ETTh1/ETTh2	transformer temperature	7	Hourly transformer measurements combining trend, periodicity, and regime-dependent variation.
	ETTm1/ETTm2	transformer temperature	7	Higher-frequency ETT benchmarks with finer local fluctuations.
	Traffic	road sensors	862	High-dimensional traffic network with recurring patterns and dense cross-sensor dependencies.
	Exchange	exchange rates	8	Financial series with weaker periodicity, nonstationary long-range behavior, and less stable correlations.

METR-LA and PEMS-BAY are traffic-network settings in the standard multivariate protocol; their spatial correlations make cross-channel information potentially useful. Solar and Electricity are dominated by periodic energy-generation or consumption patterns, but individual channels still exhibit different amplitudes and local variations. In the long-term protocol, ECL repeats the electricity

domain with a distinct preprocessing pipeline, while the ETT datasets expose trend and periodic components at different sampling frequencies. Traffic stresses both long-range temporal structure and channel interaction. Exchange is a contrasting case: its irregular and weakly periodic behavior makes small absolute metric differences produce noticeable ranking changes.

A.2 Evaluation protocols and preprocessing

Under the standard multivariate protocol, we evaluate METR-LA, PEMS-BAY, Solar, and Electricity at horizons $\{6, 24, 48, 96\}$ using R^2 and RSE. The released SeqSNN data splits, normalization, and dataset-specific input windows are retained. The long-term protocol uses a lookback window of 96 and horizons $\{96, 192, 336, 720\}$, evaluated with MSE and MAE. The complete horizon-wise table is given in Section B.

A.3 Evaluation metrics

For the standard multivariate benchmarks, the four datasets have different scales, dimensionalities, and horizon lengths. We therefore use the global scale-normalized metrics adopted by the protocol. With ground truth y_i, prediction $\widehat{y}_i$, and global mean $\bar{y}$, they are

$$R^2 = 1 - \frac{\sum_i (y_i - \widehat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}, \qquad \mathrm{RSE} = \frac{\sqrt{\sum_i (y_i - \widehat{y}_i)^2}}{\sqrt{\sum_i (y_i - \bar{y})^2}}. \tag{17}$$

This normalization makes errors comparable across datasets whose raw units and variances differ. Higher R^2 and lower RSE are better.

For long-term forecasting, we report

$$\mathrm{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \widehat{y}_i)^2, \qquad \mathrm{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \widehat{y}_i|, \tag{18}$$

where N counts all forecast entries after flattening the horizon and channel dimensions. Dataset averages are computed after averaging the four horizons within each dataset.

A.4 Implementation and reproducibility

The SeqSNN configurations use $D = 128$ for the compact channel representation. The released "spikelinear" configurations use a window of 12 for PEMS-BAY and dataset-specific windows for the remaining standard benchmarks. FSSE uses per-channel learnable decay factors and thresholds, surrogate-gradient scale 4, RevIN, component normalization, and horizon scaling. SSCA uses four attention heads, $T_s = 4$ simulation steps, query–key scale 0.125, mask rank 8, $\gamma = 1$, and mask threshold 0.3; stochastic mask sampling is disabled in both training and evaluation.

We use Adam with zero weight decay, batch size 32, learning rates specified by the released configurations, and early stopping on validation loss. Main table entries follow the configured repeated runs

B Complete Long-Term Forecasting Results

Table 5 reports all dataset–horizon combinations under the long-term protocol. Bold and underlined values denote the best and second-best entries, respectively.

The horizon-wise results reveal differences that are obscured by the dataset averages. On ECL, SpikeLite obtains the lowest MSE and MAE at all four horizons, indicating that its advantage is maintained as the prediction length increases. On Weather, its average results are comparable to SpikF, with small differences that change direction across horizons. Among the ETT datasets, SpikeLite achieves the lowest average MSE and MAE on ETTm2, whereas the results on ETTh1, ETTh2, and ETTm1 are close to those of the strongest competing methods rather than uniformly better at every horizon.

The table also clarifies where SpikeLite is less competitive. On Traffic, it improves on SpikF at every horizon but trails iTransformer, which obtains the lowest errors on this dataset. On Exchange,

Table 5: Full horizon-wise long-term forecasting results under the SpikF protocol. Each entry reports MSE and MAE. Lower values are better. Bold and underlined values denote the best and second-best results, respectively.

Dataset	Horizon	**SpikeLite**		**SpikF**		iTransformer		RLinear		PatchTST		Crossformer		TimesNet		DLinear		SCINet		Autoformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	96	.145	.236	.156	.252	.148	.240	.201	.281	.181	.270	.219	.314	.168	.272	.197	.282	.247	.345	.201	.317
	192	.161	.250	.169	.262	.162	.253	.201	.283	.188	.274	.231	.322	.184	.289	.196	.285	.257	.355	.222	.334
	336	.175	.264	.188	.281	.178	.269	.215	.298	.204	.293	.246	.337	.198	.300	.209	.301	.269	.369	.231	.338
	720	.218	.300	.219	.306	.225	.317	.257	.331	.246	.324	.280	.363	.220	.320	.245	.333	.299	.390	.254	.361
ECL Avg.		**.175**	**.263**	.183	.275	.178	.270	.219	.298	.205	.290	.244	.334	.192	.295	.212	.300	.268	.365	.227	.338
Weather	96	.158	.196	.163	.200	.174	.214	.192	.232	.177	.218	.158	.230	.172	.220	.196	.255	.221	.306	.266	.336
	192	.209	.243	.209	.241	.221	.254	.240	.271	.225	.259	.206	.277	.219	.261	.237	.296	.261	.340	.307	.367
	336	.266	.284	.266	.283	.278	.296	.292	.307	.278	.297	.272	.335	.280	.306	.283	.335	.309	.378	.359	.395
	720	.346	.337	.344	.334	.358	.347	.364	.353	.354	.348	.398	.418	.365	.359	.345	.381	.377	.427	.419	.428
Weather Avg.		**.245**	**.265**	**.245**	**.265**	.258	.278	.272	.291	.259	.281	.259	.315	.259	.287	.265	.317	.292	.363	.338	.382
ETTh1	96	.383	.391	.379	.391	.386	.405	.386	.395	.414	.419	.423	.448	.384	.402	.386	.400	.654	.599	.449	.459
	192	.437	.422	.432	.421	.441	.436	.437	.424	.460	.445	.471	.474	.436	.429	.437	.432	.719	.631	.500	.482
	336	.472	.445	.473	.441	.487	.458	.479	.446	.501	.466	.570	.546	.491	.469	.481	.459	.778	.659	.521	.496
	720	.478	.460	.474	.459	.503	.491	.481	.470	.500	.488	.653	.621	.521	.500	.519	.516	.836	.699	.514	.512
ETTh1 Avg.		.443	.429	**.440**	**.428**	.454	.447	.446	.434	.469	.454	.529	.522	.458	.450	.456	.452	.747	.647	.496	.487
ETTh2	96	.294	.336	.290	.336	.297	.349	.288	.338	.302	.348	.745	.584	.340	.374	.333	.387	.707	.621	.346	.388
	192	.366	.385	.367	.385	.380	.400	.374	.390	.388	.400	.877	.656	.402	.414	.477	.476	.860	.689	.456	.452
	336	.409	.418	.414	.420	.427	.432	.415	.426	.426	.433	1.043	.731	.452	.452	.594	.541	1.000	.744	.482	.486
	720	.426	.438	.416	.436	.427	.445	.420	.440	.431	.446	1.104	.763	.462	.468	.831	.657	1.249	.838	.515	.511
ETTh2 Avg.		.374	**.394**	**.372**	**.394**	.383	.407	.374	.398	.387	.407	.942	.684	.414	.427	.559	.515	.954	.723	.450	.459
ETTm1	96	.311	.345	.317	.345	.334	.368	.355	.376	.329	.367	.404	.426	.338	.375	.345	.372	.418	.438	.505	.475
	192	.369	.377	.372	.372	.379	.391	.391	.392	.367	.385	.450	.451	.374	.387	.380	.389	.439	.450	.553	.496
	336	.396	.395	.401	.394	.426	.420	.424	.415	.399	.410	.532	.515	.410	.411	.413	.413	.490	.485	.621	.537
	720	.473	.439	.461	.430	.491	.459	.487	.450	.454	.439	.666	.589	.478	.450	.474	.453	.595	.550	.671	.561
ETTm1 Avg.		**.387**	.389	.388	**.385**	.407	.410	.414	.407	.387	.400	.513	.496	.400	.406	.403	.407	.485	.481	.588	.517
ETTm2	96	.175	.254	.175	.251	.180	.264	.182	.265	.175	.259	.287	.366	.187	.267	.193	.292	.286	.274	.255	.339
	192	.240	.295	.242	.296	.250	.309	.246	.304	.241	.302	.414	.492	.249	.309	.284	.362	.399	.445	.281	.340
	336	.302	.336	.302	.336	.311	.348	.307	.342	.305	.343	.597	.542	.321	.351	.369	.427	.637	.591	.339	.372
	720	.395	.390	.405	.397	.412	.407	.407	.398	.402	.400	1.730	1.042	.408	.403	.554	.522	.960	.735	.433	.432
ETTm2 Avg.		**.278**	**.319**	.281	.320	.288	.327	.286	.327	.281	.326	.757	.610	.291	.333	.350	.401	.571	.537	.327	.371
Traffic	96	.453	.285	.477	.286	.395	.268	.649	.389	.462	.295	.522	.290	.593	.321	.650	.396	.788	.499	.613	.388
	192	.468	.287	.481	.289	.417	.276	.601	.366	.466	.296	.530	.293	.617	.336	.598	.370	.789	.505	.616	.382
	336	.481	.293	.499	.295	.433	.283	.609	.369	.482	.304	.558	.305	.629	.336	.605	.373	.797	.508	.622	.337
	720	.514	.311	.533	.312	.467	.302	.647	.387	.514	.322	.589	.328	.640	.350	.645	.394	.841	.523	.660	.408
Traffic Avg.		.479	.294	.497	.296	**.428**	**.282**	.626	.378	.481	.304	.550	.304	.620	.336	.625	.383	.804	.509	.628	.379
Exchange	96	.084	.204	.084	.201	.086	.206	.093	.217	.088	.205	.256	.367	.107	.234	.088	.218	.267	.396	.197	.323
	192	.180	.300	.180	.300	.177	.299	.184	.307	.176	.299	.470	.509	.226	.344	.176	.315	.351	.459	.300	.369
	336	.336	.417	.334	.417	.331	.417	.351	.432	.301	.397	1.268	.883	.367	.448	.313	.427	1.324	.853	.509	.524
	720	.864	.695	.841	.690	.847	.691	.886	.714	.901	.714	1.767	1.068	.964	.746	.839	.695	1.058	.797	1.447	.941
Exchange Avg.		.366	.404	**.360**	**.402**	.360	**.403**	.378	.417	.367	.404	.940	.707	.416	.443	**.354**	.414	.750	.626	.613	.539

the gap to SpikF is small at shorter horizons, while the larger error at horizon 720 raises SpikeLite's dataset average. These cases qualify the aggregate results in the main text: the overall performance does not imply consistent superiority on every dataset or prediction horizon.

C Stability Across Random Seeds

To assess the robustness of SpikeLite to random initialization, we repeat the experiments with three independent random seeds under both the standard multivariate and long-term forecasting protocols. For a metric value $q \in \{R^2, \mathrm{RSE}, \mathrm{MSE}, \mathrm{MAE}\}$, we report the sample mean and standard deviation across the three runs:

$$s(q) = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(q_n - \bar{q})^2}, \qquad N = 3. \tag{19}$$

Table 6 reports the three-seed results under the SpikF protocol. The "Avg." column is first computed within each run by averaging the four prediction horizons, after which the mean and standard deviation are calculated across the three runs. Table 7 provides the corresponding results under the standard multivariate protocol.

The results show that SpikeLite is stable across random initializations. Under the SpikF protocol, the standard deviations are at most 0.0012 for both MSE and MAE, and several entries remain

Table 6: Three-seed stability under the SpikF protocol. Each entry reports the mean ± sample standard deviation over three independent runs. The Avg. column is computed within each run over the four prediction horizons.

Dataset	Metric	96	192	336	720	Avg.
ECL	MSE	.1453±.0006	.1610±.0000	.1750±.0000	.2187±.0012	.1750±.0003
	MAE	.2363±.0006	.2500±.0000	.2640±.0000	.3007±.0012	.2628±.0003
Weather	MSE	.1587±.0012	.2090±.0000	.2660±.0000	.3460±.0000	.2449±.0003
	MAE	.1977±.0012	.2440±.0000	.2853±.0006	.3380±.0000	.2663±.0003
ETTm2	MSE	.1753±.0006	.2400±.0000	.3023±.0006	.3950±.0000	.2782±.0003
	MAE	.2540±.0000	.2950±.0000	.3360±.0000	.3900±.0000	.3188±.0000
Traffic	MSE	.4530±.0000	.4680±.0000	.4810±.0000	.5140±.0000	.4790±.0000
	MAE	.2850±.0000	.2870±.0000	.2930±.0000	.3110±.0000	.2940±.0000

Table 7: Three-seed stability under the standard multivariate protocol. Each entry reports the mean ± sample standard deviation over three independent runs.

Dataset	**Metric**	**Horizon 6**	**Horizon 24**	**Horizon 48**	**Horizon 96**
PEMS-BAY	$R^2\uparrow$	$.899 \pm .001$	$.781 \pm .001$	$.715 \pm .000$	$.639 \pm .003$
	$RSE\downarrow$	$.361 \pm .000$	$.506 \pm .001$	$.577 \pm .000$	$.649 \pm .002$
METR-LA	$R^2\uparrow$	$.847 \pm .001$	$.622 \pm .000$	$.452 \pm .000$	$.316 \pm .001$
	$RSE\downarrow$	$.414 \pm .001$	$.649 \pm .000$	$.781 \pm .002$	$.873 \pm .002$
Solar	$R^2\uparrow$	$.962 \pm .000$	$.885 \pm .001$	$.810 \pm .001$	$.767 \pm .001$
	$RSE\downarrow$	$.201 \pm .000$	$.348 \pm .000$	$.446 \pm .000$	$.496 \pm .000$
Electricity	$R^2\uparrow$	$.993 \pm .000$	$.991 \pm .001$	$.988 \pm .000$	$.983 \pm .000$
	$RSE\downarrow$	$.146 \pm .000$	$.167 \pm .001$	$.196 \pm .000$	$.233 \pm .000$

identical after rounding. Under the standard multivariate protocol, most standard deviations are at most 0.001, with the largest observed deviation equal to 0.003. Therefore, the main performance trends and relative rankings are not sensitive to the particular random seed.

D Additional Module Ablations

The main text focuses on Traffic because its sensor variables exhibit strong useful interactions. Table 8 gives the horizon-wise Traffic and Weather results for removing FSSE or SSCA while keeping all other settings fixed.

Table 8: Horizon-wise ablations under the long-term protocol. Lower MSE and MAE are better.

Dataset	Variant	96		192		336		720	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	SpikeLite	.453	.285	.468	.287	.481	.293	.514	.311
	w/o FSSE	.463	.291	.479	.292	.491	.298	.524	.315
	w/o SSCA	.615	.345	.581	.327	.589	.330	.624	.349
Weather	SpikeLite	.158	.197	.209	.244	.266	.285	.346	.338
	w/o FSSE	.172	.210	.223	.256	.278	.297	.358	.350
	w/o SSCA	.179	.215	.224	.256	.278	.296	.356	.357

On Traffic, removing FSSE raises the averaged MSE/MAE from $(.479, .294)$ to $(.489, .299)$, while removing SSCA produces the larger degradation to $(.602, .338)$. Weather shows smaller module gaps, consistent with weaker benefits from explicit cross-channel exchange. These results support the conditional design: FSSE supplies the input representation, while SSCA is most useful when channel interaction carries predictive information.

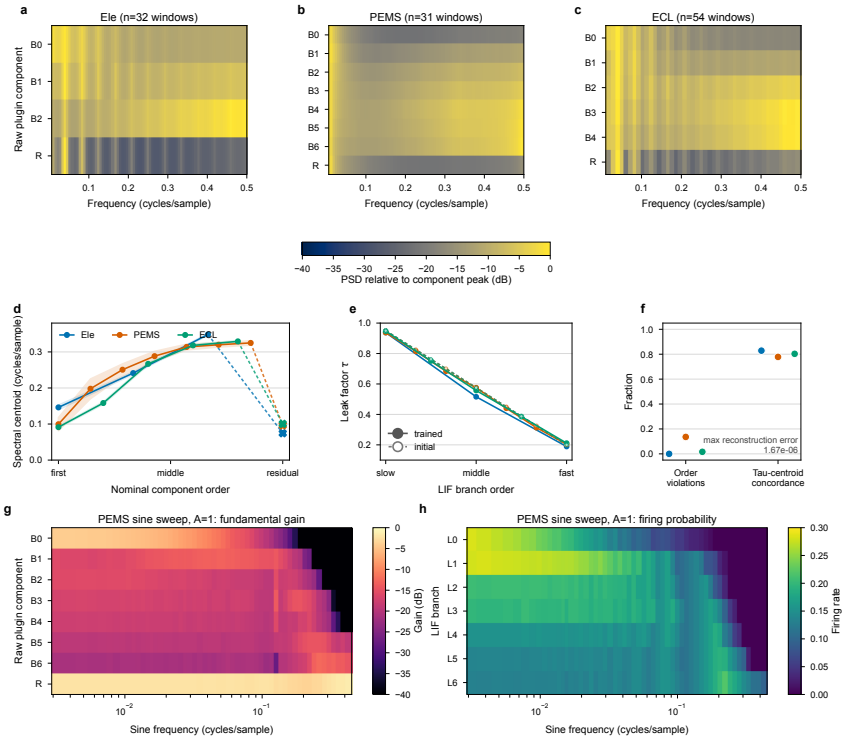


Figure 5: Frequency diagnostics for FSSE. Panels (a)–(c): real-data component spectra. Panel (d): spectral centroids. Panel (e): initialized and trained decay factors. Panel (f): branch-order violations and decay–centroid concordance. Panels (g)–(h): PEMS sinusoidal fundamental gain and firing probability.

E Frequency-Selective Mechanism Diagnostics

We analyze FSSE before its temporal projections and before SSCA. For a component $b_{n,c}^{(j)}[l]$ from test window n and channel c, we remove the temporal mean, apply a Hann window $w[l]$, and compute

$$P_{n,c}^{(j)}(f) = \left| \mathrm{rFFT}\left(w[l] \left(b_{n,c}^{(j)}[l] - \bar{b}_{n,c}^{(j)} \right) \right) \right|^2 . \tag{20}$$

The spectral centroid used in the real-data summary is

$$\mu_f^{(j)} = \frac{\sum_{f>0} f P^{(j)}(f)}{\sum_{f>0} P^{(j)}(f)} . \tag{21}$$

Figure 5 makes the panel mapping explicit: panels (a)–(c) show component spectra for Electricity, PEMS-BAY, and ECL; (d) compares component spectral centroids; (e) compares initialized and trained decay factors; (f) reports order violations and decay–centroid concordance; and (g)–(h) show the PEMS sine-sweep fundamental gain and LIF firing probability. The branches are frequency-sensitive but are not forced into disjoint bands, which is consistent with the nonlinear threshold, reset, and event-gating operations.

Before learnable component scaling and temporal projection, the successive difference construction is exactly reconstructive: $\sum_{j=0}^{K} \mathbf{B}^{(j)} = \mathbf{X}$.. The largest floating-point reconstruction error in the diagnostic is 1.67×10^{-6}. Thus, FSSE reorganizes the input response without discarding the signal at the decomposition stage.

F Energy Accounting

We report the encoder-level energy estimate for the ECL dataset with a prediction horizon of 720. We follow the established 45-nm CMOS accounting convention, assigning $E_{\mathrm{MAC}} = 4.6$ pJ to a dense multiply–accumulate operation and $E_{\mathrm{AC}} = 0.9$ pJ to an accumulation. The task-specific prediction head is excluded for every model in the comparison.

For SpikeLite, the dense operation count is decomposed into the FSSE encoder, temporal projections, and, when enabled, SSCA:

$$N_{\mathrm{dense}} = 3KLC + (K+1)CLD \tag{22}$$

$$+ \mathbb{I}_{\mathrm{SSCA}} \left[2T_s CD + T_s(3CD^2 + 2C^2D + CD^2) + N_{\mathrm{mask}} \right], \tag{23}$$

where N_{mask} accounts for the rFFT and low-rank affinity calculation. Measured spike rates are used to convert spike-driven terms into effective accumulation operations. The final energy estimate is

$$E = E_{\mathrm{MAC}} N_{\mathrm{MAC}} + E_{\mathrm{AC}} N_{\mathrm{AC}}. \tag{24}$$

G Toward Hardware Realization

SpikeLite uses hardware-compatible primitives, but the following discussion is an implementability analysis rather than a measured hardware result.

FSSE dataflow. Each input channel and LIF branch maintains one membrane state and one previous-event state. Leaky integration, thresholding, subtractive reset, and event gating are local operations with no all-to-all communication. Successive differences and the terminal residual require only additions and subtractions. Inactive events can be omitted from a compressed event stream, while an active event carries its membrane payload; this maps the encoder to a mixed event-control and membrane-value data path.

SSCA dataflow. SSCA can tile the Q/K/V projections over channels and hidden dimensions. The binary mask is generated once per input window and stored as a bit matrix or index list. Masked channel blocks can then be skipped by a scheduler during the T_s simulation steps. For large C, blockwise generation avoids materializing a dense score matrix; for dense masks, the implementation can fall back to regular tiled attention.

H Limitations and Future Work

Limitations. The current benchmarks provide regularly sampled, fully observed windows, so SpikeLite does not yet model missing values, irregular timestamps, or online arrival of observations. The reported results therefore establish the method under complete-window forecasting rather than under data-imputation or irregular-sampling conditions.

Future work. We will implement FSSE and SSCA on programmable or fabricated neuromorphic hardware and measure actual energy, latency, memory traffic, and sparse execution efficiency. We will also study irregular time series, missing-data forecasting, online prediction, and cross-dataset transfer in the future work.

I The Use of Large Language Models

Large language models were used to assist with literature retrieval, text editing, and the preparation of plotting scripts during the writing of this paper. The authors reviewed, verified, and edited all generated content and take full responsibility for the final manuscript, including all reported claims, data, and citations.